Distributed Stochastic Learning and Adaptation to Primary Traffic for Dynamic Spectrum Access

Marjan Zandi, Member, IEEE, Min Dong, Senior Member, IEEE, and Ali Grami, Senior Member, IEEE

Abstract—We design distributed online learning and channel access for secondary users in a cognitive radio network. Our goal is to design channel selection and access that can effectively adapt to a wide range of traffic load patterns in the primary network. We propose a distributed adaptive learning and access policy by applying stochastic learning automata (SLA), where each secondary user (SU) probabilistically chooses one of the most available channels to access, with the channel selection probabilities being updated based on the collision events. Our design includes two underlying distributed learning algorithms: learning of primary channel availabilities from each SU’s own sensing history, and SLA-based learning of channel selection from each SU’s own collision history for collision avoidance. We show that some existing distributed access policies can be viewed as special cases of our proposed adaptive policy, with a set of fixed channel selection probabilities. Next, we formulate the distributed channel selection and access problem as a noncooperative game. We show that it is an exact potential game with at least one pure strategy Nash equilibrium (NE). We prove that, under our proposed adaptive policy, the channel selection probabilities converge toward a pure strategy NE of the game. Simulation demonstrates the effectiveness of our proposed adaptive policy in a wide range of distributions of mean channel availabilities, as compared with other existing policies.

Index Terms—Dynamic spectrum access, stochastic learning automata, distributed channel selection and access, noncooperative game, Nash equilibrium.

I. INTRODUCTION

To overcome the inefficiency of the spectrum utilization caused by static spectrum allocation, a more intelligent and flexible spectrum allocation paradigm, namely dynamic spectrum access (DSA) using cognitive radio technology, has been proposed [2], [3]. In a hierarchical cognitive radio network, primary users, who are licensed to use the spectrum, coexist with secondary users (SUs). The SUs can only opportunistically use the licensed spectrum when channels are idle. The channel availabilities of the primary network are typically unknown to the SUs. They need to be detected by each SU through spectrum sensing. However, depending on the hardware capabilities and resource constraints, each SU can only sense a limited number of channels at any given time. With limited spectrum sensing, which channels to sense and how to make access decision directly affect the spectrum access efficiency and SU’s throughput. They are two key issues in designing dynamic spectrum access mechanisms for efficient utilization of the spectrum.

We consider a cognitive radio network consisting of N independent primary channels for licensed access and M SUs for opportunistic access. We focus on an ad hoc network setup for SUs where there is no central coordinator or dedicated control channel, and each SU independently searches available channels and makes its own access decision. Due to resource and hardware limitations for sensing, each SU can only choose one channel to sense and access in each time slot. To improve the chance for access (thus throughput), it is desirable for an SU to choose a channel that is more likely to be available. However, channel availability statistics are initially unknown to the SUs. To deal with this problem, sensing outcomes can be utilized to estimate mean channel availabilities. For distributed channel access among SUs, an effective collision resolution and/or orthogonalization mechanism is crucial to improve the throughput in the secondary network. Thus, two main issues involved in designing distributed spectrum access: How to learn the primary channel availability statistics using each SU’s own sensing observation history, and how to effectively resolve collisions and achieve self-coordination among SUs for their channel access. Moreover, the two issues are correlated with each other, making the design more challenging.

To address the issues mentioned above, several distributed learning and access policies have recently been developed [4]–[10]. In [4]–[6], the problem is formulated as a decentralized version of the classical multi-armed bandit (MAB) problem [11]–[13]. That is, each of M SUs selects, in a distributed manner, M out of N most available primary channels for access. In these policies, each SU relies on its own sensing observation history to learn the mean channel availabilities. Each policy provides a different collision avoidance or resolution mechanism for M SUs accessing the M most available channels. It has been shown that these policies are all order-optimal in terms of learning efficiency [11], meaning that the gap between the achieved throughput under these policies and that of the optimal policy grows logarithmically over time. However, when different patterns of average primary traffic loads (i.e., mean channel availabilities) across primary
channels are considered, the effectiveness of these policies is different. As analyzed in [9], between the $\rho^{\text{RAND}}$ policy [5] and the distributed learning with fairness (DLF) policy [6], the latter is more effective in performance than the former for dissimilar mean availabilities across channels, but not so for channels with similar mean availabilities.

In practice, the load conditions due to primary traffics across channels may be similar or dissimilar\(^1\). The load conditions may also be different at different times and locations. For distributed spectrum access, it is desirable to have a learning and access policy that performs well for a wide range of load conditions across channels. Towards this goal, in this paper, we aim to design a distributed learning and access policy that can automatically adjust each SU’s strategy on channel selection and access, to effectively respond to different patterns of average primary traffic loads across channels and improve SUs’ throughputs.

### A. Contributions

In this work, we consider distributed learning and channel access among $M$ SUs in a cognitive radio network through a game theoretic approach. We first propose a distributed adaptive learning and access policy which can effectively respond to different distributions of mean availability across primary channels. To do this, each SU probabilistically selects one of its estimated $M$ most available channels\(^2\) to access, and the SU updates the channel selection probabilities based on collision events. Our proposed adaptive policy is built on the following two underlying distributed learning mechanisms: 1) Learning from each SU’s own sensing history of the primary channel mean availabilities; UCB-based\(^3\) learning [14] is applied for each SU to distributively learn the mean availability of each primary channel. 2) Learning from each SU’s own collision history to adjust its channel selection among SUs for collision avoidance. We accomplish this by applying stochastic learning automata (SLA) [15], [16], where each SU tries to learn the optimum channel selection through a series of channel selections and collisions in the network.

Using these two learning algorithms in parallel, each SU’s channel selection can be adapted to effectively respond to different types of mean availability distributions across the channels. We show that both the $\rho^{\text{RAND}}$ and the DLF policies can be viewed as special cases of our proposed adaptive policy, by fixing the channel selection probabilities to specific values. Numerical results show that our proposed policy outperforms these existing policies in various types of mean channel availabilities across primary channels.

Next, we formulate the distributed channel selection and access problem as a non-cooperative strategic game. We further show that the game is an exact potential game [17], [18], and thus it can have at least a pure strategy Nash Equilibrium (NE) point. Following this, we prove that under our proposed distributed adaptive learning and access policy, the channel selection probabilities in fact converge to a pure strategy NE point of this potential game.

Simulation results show the effectiveness of our proposed algorithm in terms of throughput improvement for different types of distributions of mean availabilities across primary channels, as compared with other existing distributed learning and access policies.

### B. Related Works

As mentioned earlier, using a decentralized MAB formulation, several distributed learning and access policies have been developed [4]–[9], including the $\rho^{\text{RAND}}$ policy [5] and the DLF policy [6]. In [9], an adaptive mechanism is proposed to automatically switch between the $\rho^{\text{RAND}}$ policy and the DLF policy, as well as two different underlying learning policies, by estimating the distribution type of the mean availabilities across primary channels (e.g., similar or dissimilar). However, determining the switching threshold can be tricky, and the mechanism there is designed only to switch between the two aforementioned policies.

Various game approaches have been considered for designing channel selection and access policies in cognitive radio networks [19]–[23], where SUs’ accesses have been modeled and formulated using a certain type of games. In [19], pricing-based DSA mechanisms that enable SUs to contend for channel usage are devised. In [20], decentralized DSA where each SU has noisy measurements of all primary channels is studied using the theory of multivariate global games. The conditions ensuring the Bayesian NE are derived. In [21], the competitions of the SUs are modeled as singleton congestion games, and the difference between NE and social optimality is evaluated through studying the price of anarchy. In [22], the problem of the SUs decision making process is formulated as a Chinese Restaurant Game by considering the scenario where SUs sense channels simultaneously and make access decisions sequentially. In [23], a hierarchical game using a network coordinator is proposed and a Stackelberg game is applied where the network coordinator acts as the leader and SUs act as the followers. None of the above works investigate the effect of relative primary channel availabilities distributions across channels on the efficiency of the game performance.

SLA is used as a learning model to adapt decision making in an unknown environment [15], [16]. In [16], a framework of decentralized learning of NE in a multi-player stochastic game is provided. An SLA-based algorithm is proposed and its NE achieving properties are analyzed. SLA has been applied in a wide range of problems in different areas [10], [24]–[26]. For example, in [25], the SLA is incorporated in designing probabilistic power adaptation algorithms. In [26], stochastic automata rate adaptation algorithm is proposed which implements the SLA in the context of rate adaptation. In the DSA context, an SLA-based distributed access policy is recently proposed in [10]. It applies a direct learning automata\(^4\) named

---

1. As an example, for a licensed cellular spectrum, the channels are assigned to different primary users who may have different types of traffic and usage intensity, e.g., voice, FTP, or video streaming. Thus, the traffic loads on different channels may exhibit different statistics.

2. As it will be clear later, the $M$ most available channels refer to those channels with $M$ highest channel mean availabilities.

3. UCB stands for upper-confidence-bound.

4. A direct learning automata is a learning automata in which the environment model is not used in the learning algorithm.
Linear Reward-Inaction algorithm [15]. A different mechanism for channel selection and collision resolution is designed. No learning of the channel availabilities is performed at SUs, and thus the selection does not distinguish channels with different mean availability statistics. Through simulation, we show that our proposed policy outperforms this policy in various types of mean availabilities across primary channels.

Besides the above mentioned results, various auction-based approaches have also been proposed for efficient spectrum access, sharing, or leasing [27]–[33]. These approaches often require a network coordinator (or auctioneer), and thus belong to the type of centralized approaches for channel selection and access design.

C. Organization and Notations

The remainder of this paper is organized as follows. Section II describes the network model. In Section III, we provide our proposed distributed adaptive learning and access policy in which the underlying learning algorithms adapt the channel selection according to the mean availability distribution across channels. In Section IV, we formulate this distributed channel selection and access problem as an exact potential game. In Section V, we show that our proposed policy converges to the pure strategy NE of this game. In Section VI, we provide some discussions on the network model extension and practical implementation issues. Section VII provides simulation studies and comparisons with existing policies, and Section VIII concludes the paper.

Notations: The main symbols used in this paper are summarized in Table I.

### TABLE I

**The List of Main Notations**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>number of secondary users</td>
</tr>
<tr>
<td>$N$</td>
<td>number of channels</td>
</tr>
<tr>
<td>$n$</td>
<td>current time slot</td>
</tr>
<tr>
<td>$X_i(n)$</td>
<td>availability state of the primary channel $i$ at time slot $n$</td>
</tr>
<tr>
<td>$\theta_i$</td>
<td>mean availability of channel $i$</td>
</tr>
<tr>
<td>$\hat{\theta}_i$</td>
<td>estimated mean availability of channel $i$ by SU $j$</td>
</tr>
<tr>
<td>$T_i^j(n)$</td>
<td>number of times that the SU $j$ senses channel $i$ up to time slot $n$</td>
</tr>
<tr>
<td>$S_i^j(n)$</td>
<td>number of time slots up to the current time slot $n$ that SU $j$ has been the sole user of channel $i$</td>
</tr>
<tr>
<td>$R(n)$</td>
<td>regret</td>
</tr>
<tr>
<td>$V_j(n)$</td>
<td>index vector for SU $j$ based on its own sensing observation history</td>
</tr>
<tr>
<td>$I_i^j(n)$</td>
<td>rank of channel $i$ at time slot $n$ for SU $j$</td>
</tr>
<tr>
<td>$c_{i,M}(n)$</td>
<td>set of indexes of the estimated $M$-best channels of SU $j$ at time slot $n$</td>
</tr>
<tr>
<td>$\sigma_{j,r}$</td>
<td>index of the channel who is ranked $r$th in $V_j(n)$ for SU $j$</td>
</tr>
<tr>
<td>$p_i^j(n)$</td>
<td>probability of selecting channel $i$ by SU $j$ at time slot $n$</td>
</tr>
<tr>
<td>$C_i^j(n)$</td>
<td>acknowledgment of SU $j$’s collision on channel $i$</td>
</tr>
<tr>
<td>$T_i^j(n)$</td>
<td>reward received by SU $j$ by accessing the primary network at time slot $n$</td>
</tr>
</tbody>
</table>

II. Network Model

We consider a slotted primary network consisting of $N$ radio channels, and a secondary network of $M$ SUs independently searching opportunities for idle channels in the primary network. We assume that these SUs are in a close range, and thus they interfere with each other when simultaneously transmitting over the same channel. We assume $M$ is known to the SUs. We are interested in the scenario where $M < N$. This scenario arises in broadband spectrum access where there are a large number of primary channels for consideration.

As we will see, our developed policy also applies to the scenario where $M \geq N$. However, the scenario where $M < N$ is more challenging, because there is a nontrivial learning process for efficient access to maximize SUs’ throughputs, as we will see in Section III.

Let $X_i(n)$ denote the availability state of channel $i$ in the primary network at time slot $n$, where $X_i(n) = 1$ if channel $i$ is available and 0 otherwise. We assume that $X_i(n)$ evolves as a stationary Bernoulli random process over $n$, with the mean $\theta_i = E[X_i(n)] \in [0, 1]$, i.e., $X_i(n) \sim$ Bernoulli($\theta_i$), $\forall n$ and $i = 1, \cdots, N$. We assume $\theta_i$’s are distinct from each other and are unknown to the SUs. Denote $\theta = [\theta_1, \theta_2, \cdots, \theta_N]$ as the mean channel availability vector.

We assume that all SUs are active, i.e., they always have data to send and are searching for idle channels to access in each time slot. We focus on the scenario where SUs sense and access the channels without exchanging local information. Due to hardware limitation, we assume each SU can sense and access one channel in each time slot. Specifically, at the beginning of time slot $n$, each SU selects a channel to sense and access if available. In this work, we assume perfect channel sensing at all SUs. Each SU tries to learn the unknown mean availability statistics. Through simulation, we show that our developed policy outperforms this policy in various types of mean availabilities across primary channels.

Regret is a common metric used to measure the throughput loss of a given access policy under learning [11]. It is defined as the difference of throughput between the ideal scenario and a given policy. The ideal scenario is when $\theta$ is known to the SUs. The maximum network throughput in this case, denoted by $R^*(n)$, is given by $R^*(n) = n \sum_{k=1}^{M} \theta_k$, where $k^*$ represents a channel index such that $\theta_k^*$ is the $k$th-highest valued element in $\theta$. Thus, by definition, regret is given by

$$R(n) \overset{\Delta}{=} R^*(n) - \sum_{i=1}^{N} \sum_{j=1}^{M} \theta_i E[S_i^j(n)]$$

This scenario is particularly suitable for the cognitive radio network where the SUs may not be restricted to a certain frequency band and can search among a large set of channels.
where \( S_j^i (n) \) denotes the number of times, up to current slot \( n \), that SU \( j \) is the sole user to sense channel \( i \).

Our design objective is to devise a distributed access policy aiming at optimizing each SU’s own performance and minimizing the regret. In this case, no network coordinator is required and each SU makes its own sensing and access decision.

### III. A Distributed Adaptive Learning and Access Policy

In this section, we first describe our proposed distributed adaptive learning and access policy which is built on two underlying learning algorithms. In Section V, we will show that under this policy, the channel selection solution will converge to the NE of the game formulated in Section IV for distributed channel selection and access.

#### A. Distributed Learning and Sensing of Primary Channels

When the availabilities across channels are different, it is desirable for an SU to choose the most likely available channel to access. However, the mean channel availability vector \( \theta \) is unknown. A sensing policy is designed to learn the unknown \( \theta \). In [14], the upper-confidence-bound 1 (UCB1), an online learning algorithm of the unknown parameters (e.g., \( \theta_i \)’s), is proposed for a single player MAB problem with \( N \) arms (e.g., channels). It is an index-based learning policy, in which an index is computed for each arm and is used to rank the arm. At each time slot \( n \), the player then selects the highest-ranked arm. In [11], [14], it is shown that, if the random reward process in each arm (e.g., \( X_i (n) \)) is i.i.d., then the UCB1 is an order-optimal learning algorithm in the sense that the algorithm achieves the logarithmic growth of the regret over time.

For dynamics spectrum access (DSA), decentralized access policies proposed in [4]–[6] extend the UCB1 algorithm to the distributed case for the underlying learning of channel availability statistics. Here, we also adopt the UCB1 algorithm for distributed learning of \( \theta \). Let \( T_i^j (n) \) denote the number of times that SU \( j \) senses channel \( i \) up to time slot \( n \). If SU \( j \) selects channel \( i \) to sense at time slot \( n \), then it obtains the value of \( X_i (n) \) and records it as \( X_i^j (T_i^j (n)) \). Let \( X_i^j (n) = [X_i^j (1), \cdots, X_i^j (T_i^j (n))] \) be the vector holding the sensing observation history of channel \( i \) up to time slot \( n \) at SU \( j \). Using \( X_i^j (n) \), SU \( j \) estimates \( \theta_i \) of channel \( i \) at time \( n \) as

\[
\hat{\theta}_i^j (T_i^j (n)) \triangleq \frac{1}{T_i^j (n)} \sum_{k=1}^{T_i^j (n)} X_i^j (k). \tag{2}
\]

Extending the UCB1 algorithm in [14] to the case of distributed users, we define a ranking-index for channel \( i \) at SU \( j \) as

\[
I_i^j (n) \triangleq \hat{\theta}_i^j (T_i^j (n)) + \sqrt{\frac{2 \log n}{T_i^j (n)}}, \tag{3}
\]

Each arm provides a random reward from a distribution with an unknown parameter specific to that arm.

It will be used for ranking the channels by SU \( j \). Specifically, each SU \( j \) computes a ranking-index vector \( I^j (n) = [I_1^j (n), \cdots, I_M^j (n)] \) based on its own observation history. Then, it selects the channel whose index value is the \( k \)-th highest in \( I^j (n) \) to access, for some \( 0 \leq k \leq M \).

**Remark 1:** Note that the two terms in (3) capture the exploration and exploitation trade-off in the learning process. The sample mean for the estimated channel availably \( \theta_i \) in the first term corresponds to exploitation, while the second term is used for exploration which adds weights to those channels that are not sensed often. Therefore, the trade-off is between choosing a channel with a high estimated availability for immediate throughput maximization and choosing another channel to obtain an improved estimate of its availability.

**Remark 2:** As mentioned earlier, for a single player MAB problem, the UCB1 algorithm is an order-optimal learning algorithm for i.i.d. random reward processes. Thus, if the underlying primary channel availability \( X_i (n) \) is i.i.d. over time, then the distributed learning process described above is efficient in reducing regret and improving throughput. This i.i.d. assumption is used in existing distributed learning policies [4]–[6].


As shown in (1), in the ideal scenario where all \( \theta_i \)’s are known to the \( M \) SUs, the strategy to achieve the highest sum throughput is to let the SUs orthogonally choose among the \( M \) channels with the \( M \) highest \( \theta_i \)’s. According to this, existing distributed learning policies [4]–[6] propose different mechanisms for access coordination among SUs to choose different channels among the first \( M \)-highest ranked channels at each SU. As shown in [9], these policies may work well in one type of mean channel availability distribution in \( \theta \) but not in other cases. In other words, the effectiveness of a proposed access policy will be impacted by different channel availability distributions, resulting in different relative throughput performances. Designing learning and access policies that work well for a wide range of mean channel availability distribution \( \theta \) of the primary channels is practically desirable, and is our goal in this work.

To design such a policy, we propose an adaptive learning and access policy based on the idea of SLA [15]. By applying SLA, each SU adapts its channel selection through learning from its collision history. SLA is used as a learning model in an unknown random environment. For spectrum access, the primary network can be considered as the unknown random environment. Each SU is considered as a learning automaton for adaptive decision making. Specifically, we let each SU distributively estimate the set of the \( M \) most available channels, and probabilistically select one channel from this set to access. The channel selection probabilities at each SU is adapted over time by using SLA based on collision history. In Section V, we show that over the long run, our algorithm leads to orthogonalization among SUs’ channel selections.

Define the \( M \)-best channels as those channels with \( \theta_i \) values being among the \( M \) highest ones in \( \theta \), and the estimated
M-best channels as those channels whose ranking-index $I_j^i(n)$’s are among the $M$ highest in $I^i(1)$. Denote $\sigma_{j,r}$ the index of the channel who is ranked $r$th in $I^i(1)$ for SU $j$. Let $\mathcal{C}_M^j(n)$ be the set of indexes of the estimated M-best channels for SU $j$ at time slot $n$, given by

$$\mathcal{C}_M^j(n) = \{\sigma_{j,1}(n), \ldots, \sigma_{j,M}(n)\}. \quad (4)$$

We let each SU $j$ probabilistically choose a channel for sensing. Let $p_j^i(n)$ denote the probability of selecting channel $i$ by SU $j$ at time slot $n$. Let the channel selection probability distribution vector for SU $j$ by

$$\mathbf{p}_j^i(n) = [p_j^i(n), \ldots, p_j^N(n)]^T \quad (5)$$

where entries of $\mathbf{p}_j^i(n)$ satisfies $\sum_{i=1}^N p_j^i(n) = 1$. At the start of the process, SU $j$’s channel selection probability distribution is uniformly initialized to $\mathbf{p}_j^i(0) = [1/N, \ldots, 1/N]$.

1) Channel Selection: At time slot $n$, SU $j$ selects a channel in the set of its estimated M-best channels $\mathcal{C}_M^j(n)$. To do this, we re-normalize the channel selection probabilities of these channels, i.e., $p_j^i$, for $i \in \mathcal{C}_M^j(n)$. Define $\tilde{p}_j^i(n)$ as the re-normalized probability of the channel $i$, for $i \in \mathcal{C}_M^j(n)$. It is computed as

$$\tilde{p}_j^i(n) = \frac{p_j^i(n)}{\sum_{i \in \mathcal{C}_M^j(n)} p_j^i(n)}, \quad \text{for } i \in \mathcal{C}_M^j(n). \quad (6)$$

Let $\mathbf{\tilde{p}}_j^i(\tilde{n}) = [\tilde{p}_j^{i,1}(\tilde{n}), \ldots, \tilde{p}_j^{i,M}(\tilde{n})]$ be the re-normalized channel selection probability vector of the estimated M-best channels for SU $j$. It then selects the rank $r \in \{1, \ldots, M\}$ with the probability distribution $\mathbf{\tilde{p}}_j^i(\tilde{n})$, and chooses the corresponding channel with index $\sigma_{j,r}$, i.e., the $r$th-highest ranked channel in $\mathcal{C}_M^j(n)$ based on the ranking-index vector $I^i(1)$. SU $j$ then senses the selected channel and accesses it if available.

In the next time slot $(n + 1)$, SU $j$ updates its estimated M-best channel set $\mathcal{C}_M^j(n + 1)$ and its ranking-index vector $I^i(n + 1)$. If no collision occurs, SU $j$ will maintain its previous rank selection $r$, and select a channel with index $\sigma_{j,r}$ from $\mathcal{C}_M^j(n + 1)$. Otherwise, SU $j$ will redraw the rank $r \in \{1, \ldots, M\}$ with $\mathbf{\tilde{p}}_j^i(\tilde{n} + 1)$, and select channel $\sigma_{j,r}$.

2) Channel Selection Probability Update: Each SU $j$ uses an acknowledgement for collision feedback. Let $\zeta_j^i(n) \in \{0, 1\}$ denote the acknowledgment of SU $j$’s collision on channel $i$, where $\zeta_j^i(n) = 1$ denotes the collision event and 0 otherwise. Based on the collision model, for SU $j$, we define $\Upsilon_j^i(n)$ as the reward in terms of throughput by accessing channel $i$ at time slot $n$, given by

$$\Upsilon_j^i(n) = \begin{cases} 1, & \text{if } X_j(n) = 1, i = \sigma_{j,r}(n), \text{and } \zeta_j^i(n) = 0, \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

Define $\Upsilon_j^i(n)$ as the reward received by SU $j$ by accessing the primary network at time slot $n$. Since each SU at most selects one channel to access at a time, we have $\Upsilon_j^i(n) \in \{0, 1\}$ which is given by

$$\Upsilon_j^i(n) = \sum_{i \in \mathcal{C}_M^j(n)} \Upsilon_j^i(n) = \Upsilon_j^i(\sigma_{j,r}(n)). \quad (8)$$

Due to possible collision, $\Upsilon_j^i(n)$ is random.

Based on the reward $\Upsilon_j^i(n)$, SU $j$ updates its channel selection probability distribution $\mathbf{p}_j^i(n)$ to $\mathbf{p}_j^i(n + 1)$ according to the following SLA-based rule:

$$p_j^i(n + 1) = p_j^i(n) + b \Upsilon_j^i(n)(1 - p_j^i(n)), \quad \text{for } i = \sigma_{j,r}(n) \quad (9)$$

$$p_j^i(n + 1) = p_j^i(n) - b \Upsilon_j^i(n)p_j^i(n), \quad \text{for } i \neq \sigma_{j,r}(n) \quad (10)$$

where $b \in (0, 1)$ is the updating step size. From (8)-(10), we verify that $\sum_{i=1}^N p_j^i(n + 1) = 1$. Note that the above SLA-based updating rule adaptively adjusts the channel selection probabilities based on the reward from the access attempt. When $\Upsilon_j^i(n) = 0$ (either due to the primary channel not being available or collision with other SUs), $p_j^i(n) = p_j^i(n + 1), \forall i$. When SU $j$ accessing channel $\sigma_{j,r}$ is successful, $p_j^i(n + 1)$ will be increased, for the accessed channel $i = \sigma_{j,r}$, while that for the rest channels will be decreased.

A summary of the proposed distributed adaptive learning and access (DALA) policy is provided in Algorithm 1.

Remark 1: The value of the updating step size $b$ in (9) and (10) affects the convergence speed, as well as the variation of channel selection probability $p_j^i(n)$ over time. A smaller value of $b$ (e.g., $b \ll 1$) leads to less variation of $p_j^i(n)$ over time, but slower the convergence speed, while a larger value of $b$ leads to faster convergence but a more significant variation of $p_j^i(n)$ over time. In addition, if the value of $b$ is too large, the procedure may not guarantee to converge. The preferred value of $b$ can be determined through experimental testing.

Remark 2: In our proposed policy, when a collision occurs, in the next time slot, the affected SU transmitter probabilistically redraws the rank to select a channel for sensing and access. In this case, the SU transmitter and receiver pair needs to synchronize the channel selection. For an ad hoc secondary network, the SU transmitter needs to inform the SU receiver of the selected channel. Depending on the specific protocol structure and channelization design in the secondary network, this can be accomplished in various ways. For example, a dedicated control channel can be used for the related control information (e.g., channel selection, packet acknowledgement, etc.). The SU transmitter can use this control channel to inform the SU receiver of its channel selection.

Remark 3: As mentioned in Section II, we are mainly interested in the scenario where $M < N$ in designing the learning and access policy. In this case, in order to minimize the regret and improve the network throughput, there is a nontrivial learning process for each of $M$ SUs to determine the $M$-best
Algorithm 1. Distributed adaptive learning and access (DALA) policy: //For SU $j$ at time slot $n$

1: Input:
   $N$: Number of primary channels;
   $M$: Number of secondary users;
   $n$: Current time slot;
   $r$: The rank of channel used at time slot $n - 1$;
   $\zeta_{j}^{r}(n - 1) \in \{0, 1\}$: The collision acknowledgment for user $j$ on channel $i$.
   $p_{th}$: Threshold for checking convergence of $p_{j}^{r}(n)$
2: If $n == 0$ then //initialization
   Set $p_{j}^{r}(n) = \left\{ \frac{1}{M}, \ldots, \frac{1}{M} \right\}$.
end if
3: Select channel to sense and access:
   i) Obtain the ranking vector $V_{j}(n)$ using (2) and (3).
   ii) Update $\hat{\theta}_{j}^{r}(n)$ in (4).
   iii) If $\zeta_{j}^{r}(n - 1) == 1$ then //collision
         Redraw $r \sim \tilde{\theta}^{r}(n)$ as in (6).
      else //no collision
         Keep its previous rank $r$.
   end if
   iv) Select channel $\sigma_{j,r}$ from $C_{j}^{r}(n)$ for sensing; Obtain sensing result $X_{\sigma_{j,r}}(n)$.
      If $X_{\sigma_{j,r}}(n) == 1$ then //channel is available
         Access channel $\sigma_{j,r}$.
      end if
   v) Update acknowledgement:
      If collision then $\zeta_{\sigma_{j,r}}^{r}(n) = 1$.
      else $\zeta_{\sigma_{j,r}}^{r}(n) = 0$.
   end if
4: Update channel selection probability vector:
   i) Obtain the reward $Y_{j}(n)$ as in (12).
   ii) Update $p_{j}^{r}(n + 1)$ according to (9) and (10).

Our proposed channel sensing and access policy consists of two underlying distributed learning algorithms:
1) Distributed learning of the primary channel mean availability $\theta$ based on each SU’s own sensing history. This learning mechanism ensures that each SU selects a channel among the most available channels for access to maximize its throughput.
2) Distributed learning of channel selections among SUs through collision events to automatically adjust and orthogonalize each SU’s channel selection. This learning is reflected in the converging value of $p_{j}^{r}(n)$ over time for each SU $j$. As it will be evident in simulation, over time, each SU $j$ will eventually select a particular channel with probability approaching 1. Furthermore, the channel selections among SUs are orthogonalized.

In fact, the two learning processes are intertwined. With a closer examination of the learning of channel selections, we observe that the update of channel selection probability in (9) and (10) is impacted by the accuracy of the learning of $\theta$, i.e., the estimate $\hat{\theta}^{r}$. For the values of $\theta$’s being spread out, the learning of $\theta$ is relatively more accurate over time, especially for those $M$-best channels (being chosen more often to sense and thus more samples). This means that for SU $j$, a channel $i$’s rank in the ranking-index vector $V_{j}(n)$ will match more accurately to its actual rank in terms of $\theta_{i}$. Therefore, SU $j$’s estimated $M$-best channel set $C_{j}^{r}(n)$ will be close to the actual $M$-best channel set. Consequently, if no collision happens, SU $j$’s rank selection $r_{j}$ and channel selection $\sigma_{j}$ will mostly remain unchanged. This results in a quick convergence of channel selection probability $p_{j}^{r}(n)$ over time to a pure probability vector of a specific rank $r_{j}$ (in other words, selecting a rank $r_{j}$ with probability approaching 1). As a result, the access policy effectively converges to a policy in which each SU selects a channel with a fixed (orthogonalized) rank to access.

On the other hand, if channel availabilities are similar, i.e., values of $\theta_{i}$ being similar, the learning of $\theta$ is relatively inaccurate and slow over time. Specifically, SU $j$’s estimated $M$-best channel set $C_{j}^{r}(n)$ varies over time and does not closely match the actual $M$-best channels. In this case, each SU may rank channels differently, due to the mismatch of rank $r_{j}$ and the true rank of the channel availability. This results in a more collision-prone scenario among SUs, and in return, a re-selection of the rank $r_{j}$ for SU $j$, based on Algorithm 1. Consequently, $p_{j}^{r}(n)$ will change slowly from the initial uniform distribution before the estimate $\hat{\theta}^{r}$ becomes more accurate. The benefit of this slow convergence is that during this process, if collision happens, each SU actively re-selects a channel to access, thus proactively resolving collision among users to reduce the throughput loss. During this period, using a pure probability vector instead at each SU may result in more collisions.

Thus, as we see, through such an adaptive change of channel selection probability vector, our proposed algorithm nicely adjusts its “collision resolution strategy” based on the mean channel availability distribution $\theta$ across primary channels.

D. Relation to Existing Distributed Access Policies

The $\rho^{\text{RAND}}$ policy [5] and the DLF policy [6] are two existing MAB-based distributed access policies which use different mechanisms for access coordination. The access mechanisms ensure that SUs select different channels among the estimated $M$-best channels. Both of the policies have been shown to be order-optimal in terms of the growth rate of regret. They are briefly described below:
1) $\rho^{\text{RAND}}$ policy [5]: Each SU $j$ selects a random rank $r_{j}$ uniformly from 1 to $M$. It will then access the channel $i$ whose $I_{j}^{r}(n)$ is ranked $r_{j}^{th}$ in $V_{j}(n)$. At time slot $n$, if a collision
occurred in the previous slot, SU \( j \) will re-draw \( r_j \) again; otherwise, it keeps the previously generated rank \( r_j \) for channel selection.

2) DLF policy \([6]\): At time slot \( n \), SU \( j \) selects the \( r_j^th \)-rank channel to access among the top \( M \)-ranked channels in terms of \( \Psi(n) \), where the rank \( r_j \) for each SU is generated in a round robin fashion \( r_j = ((j + n) \mod M) + 1 \).

Based on our analysis in Section III-C, we observe that both \( \rho^{\text{RAND}} \) and DLF can be considered as a special case of our proposed DALA policy in Algorithm 1, with the channel selection probability distribution \( \tilde{\mathbf{p}}(n) \) of estimated \( M \)-best channels being either uniform or pure (i.e. ‘0’ or ‘1’ value).

The \( \rho^{\text{RAND}} \) and DLF policies are analyzed and numerically compared in [9], and it is observed that they perform differently for different \( \theta \) distribution. The \( \rho^{\text{RAND}} \) policy performs better when the values of \( \theta_i \)'s are similar, and the DLF policy performs better when the values of \( \theta_i \)'s are relatively spread out. Our proposed DALA policy provides an adaptive approach to respond to different \( \theta \) distribution. As will be shown in our simulation, our proposed DALA policy outperforms both two policies in a wide range of \( \theta \) distributions.

IV. GAME THEORETIC FORMULATION

In the following, we formulate the distributed channel selection and access problem previously considered as a non-cooperative strategic game\(^8\) [34], where the SUs are considered as the players, and the licensed primary channels are the possible actions that the SUs may take.

Specifically, let \( \mathcal{S} = \{1, \ldots, M\} \) denote the set of SUs and \( \mathcal{C} = \{1, \ldots, N\} \) denote the set of channels for the SUs to select (actions); the action can be a primary channel \( i \in \{1, \ldots, N\} \). Let \( \mathcal{A}_j(n) \subseteq \mathcal{C} \) denote the set of channels that SU \( j \) can select from at time slot \( n \) and \( a_j(n) \) denote the channel selection (action) taken by SU \( j \) at time slot \( n \). Define \( a_{-j}(n) \) as the set of channel selections taken by SU \( j \)'s opponents as \( a_{-j}(n) \Deltaq \{a_j(n) : j' \notin \{j\} \} \). Denote \( u_j(a_j, a_{-j}; n) \) the payoff of SU \( j \) upon taking action \( a_j(n) \in \mathcal{A}_j(n) \) while others taking \( a_{-j}(n) \). Then, \( \mathcal{S}_p = \{\mathcal{S}, \{\mathcal{A}_j(n)\}_{j \in \mathcal{S}}, \{u_j(a_j, a_{-j}; n)\}_{j \in \mathcal{S}}\} \) forms a game [35]. Note that after each SU \( j \) determines its action \( a_j(n) = q \), it will make access decision depending on the channel availability, i.e. \( X_q = 1 \).

Let \( m_a(n) \) denote the number of SUs taking the same action \( a(n) \) at time slot \( n \). Based on SU \( j \)'s acknowledgement \( \zeta_d(n) \) defined in Section III-B2, we have the following equivalence

\[
\begin{align*}
  m_a(n) = 1 & \iff \zeta_d(n) = 0, \\
  m_a(n) > 1 & \iff \zeta_d(n) = 1.
\end{align*}
\]

We set the action set for SU \( j \) as \( \mathcal{A}_j(n) = \mathcal{C}_M^j(n) \), i.e., SU \( j \) can select one of its estimated \( M \)-best channels. Then, accessing channel \( i \) (i.e., \( i = \sigma_j(n) \)) can be viewed as taking action \( a_j(n) \) where \( X_i = 1 \). Therefore, the reward \( \Upsilon_d(n) \) in (7) can be interpreted as the reward for SU \( j \) taking action \( a \), re-expressed as

\[
\Upsilon_d(n) = \begin{cases} 
  1, & \text{if } a(n) = a_j(n), m_a(n) = 1, \text{ and } X_{a_j}(n) = 1; \\
  0, & \text{otherwise}.
\end{cases}
\] 

Similarly, the reward \( \Upsilon_d(n) \) in (8) can be re-expressed by

\[
\Upsilon_d(n) = \sum_{a \in \mathcal{A}_j(n)} \Upsilon_d(n) = \Upsilon_d(n) = \Upsilon_d(n).
\]

We define the payoff of SU \( j \) as its expected throughput achieved from accessing the primary network, defined as

\[
u_j(a_j, a_{-j}; n) \triangleq E[\Upsilon_d(n)|a_j(n), a_{-j}] = \psi_{a_j}(m_{a_j}(n))
\]

where the expectation is taken over the randomness of the channel availability state, and for \( a_j(n) \in \mathcal{A}_j(n) \),

\[
\psi_{a_j}(m_{a_j}(n)) = \begin{cases} 
  \theta_{a_j}, & m_{a_j}(n) = 1; \\
  0, & \text{otherwise}.
\end{cases}
\]

A. Exact Potential Game

A game is called a potential game \([17], [18]\), where the incentives of all players of the game for changing their actions can be reflected by a function which is called a potential function. Showing the existence of a potential function in a game is sufficient to prove the game being a potential game.

Let \( \mathcal{P}(a_j, a_{-j}; n) \) denote the potential function of a game if SU \( j \) takes action \( a_j(n) \in \mathcal{A}_j(n) \) and \( \mathcal{P}(\tilde{a}_j, a_{-j}; n) \) denote the potential function if SU \( j \) takes action \( \tilde{a}_j(n) \in \mathcal{A}_j(n) \). An exact potential game \([17], [18]\) is defined as a game where there exists a potential function such that if player \( j \) changes its action from \( a_j(n) \) to \( \tilde{a}_j(n) \), the deviation in the payoff of player \( j \) is reflected by deviation in the potential function, i.e.

\[
\mathcal{P}(\tilde{a}_j, a_{-j}; n) - \mathcal{P}(a_j, a_{-j}; n) = u_j(\tilde{a}_j, a_{-j}; n) - u_j(a_j, a_{-j}; n)
\] 

where \( a_j(n) \in \mathcal{A}_j(n) \) and \( a_{-j}(n) \in \mathcal{A}_{-j}(n) \).

1) Property of NE: One of the most important properties of the exact potential game is that it can achieve at least one pure-strategy NE \([17]\).

Pure strategy: In a stochastic game, a player selects an action with certain probability. The channel selection probability distribution vector \( \mathbf{p}^d(n) \) below (6) defines a strategy for SU \( j \). Any unit probability distribution vector (i.e., only one entry with 1 and the rest 0) represents a pure strategy. Other non-unit vectors represent mixed strategies.

Pure-strategy NE: At time slot \( n \), an action profile \( a^o(n) = [a^o(n) \in A_M, \ldots, a^o_M(n)] \) is called a pure strategy NE for \( \mathcal{S}_p \), if and

\[^8\] A strategic game is a game in which the interaction of the players (decision-makers) are considered where each decision-maker selects its action once, and these actions are taken simultaneously.

\[^9\] In other words, player \( j \) taking an action \( a^o_j(n) \in a^o(n) \) with probability 1.
only if no SU can obtain a higher payoff by deviating unilaterally from this profile. That is, let \( m_a(n) \) and \( m_j(n) \) be the number of SUs selecting actions \( a_j(n) \) and \( a_j(n) \), respectively, at time slot \( n \). If SU \( j \) changes its action from \( a_j(n) \) to \( a_j(n) \) while others keep their actions unchanged, the number of SUs selecting actions \( a_j(n) \) becomes \( m_a(n) + 1 \). Then, the following property holds
\[
\psi_a(m_a(n)) \leq \psi_a(m_a(n) + 1),
\]
where \( \forall a_j(n) \in A_j(n) \setminus \{a_j(n)\} \), \( \forall j \in S. \) (16)

**B. \( \mathcal{S}_p \) as an Exact Potential Game**

We now show that with the payoff defined in (13), there exists a potential function with the property as in (15), and the game \( \mathcal{S}_p \) is an exact potential game.

**Proposition 1:** The channel selection and access game \( \mathcal{S}_p \) is an exact potential game.

**Proof:** See Appendix A.

By Proposition 1 and the property of the exact potential game, it follows that the game \( \mathcal{S}_p \) can at least achieve one pure strategy NE point. In the next section, we analyze the convergence behavior of our proposed algorithm in Section III-B to the NE points of the game \( \mathcal{S}_p \).

**V. ACHIEVING PURE STRATEGY NE OF THE GAME \( \mathcal{S}_p \)**

In this section, we show that under our proposed DALA policy (Algorithm 1) in Section III, the channel selection probability vector converges to the pure strategy NE of the game \( \mathcal{S}_p \).

A multi-person stochastic game is considered in [16], where the convergence of a proposed SLA-based algorithm to NE is analyzed. Although our game \( \mathcal{S}_p \) is different from the game defined there, we can adopt the approach in [16] to investigate the convergence of our algorithm. That is, we use the solution of ordinary differential equation (ODE) to analyze and understand the long term behavior of the channel selection probability matrix \( P(n) \).

Define \( P(n) \equiv \{p^1(n), p^2(n), \ldots, p^M(n)\} \) as the \( N \times M \) channel selection probability matrix, where \( p^i(n) \) is given in (5). Define \( a(n) \equiv \{a_1(n), a_2(n), \ldots, a_M(n)\} \) as the channel selection (action) profile and \( Y(n) \equiv \{Y^1(n), Y^2(n), \ldots, Y^M(n)\} \) the reward profile at time slot \( n \). Then, (9) and (10) can be combined to be re-expressed as
\[
P(n + 1) = P(n) + bG(P(n), a(n), Y(n)) \tag{17}
\]
where \( G(P(n), a(n), Y(n)) \) is specified by the updating rule in (9) and (10), and is a function of \( P(n), a(n), \) and \( Y(n) \). Define \( f(P) \) as the conditional expectation of \( G(\cdot) \) given \( P(n) = P \) as
\[
f(P) = E[G(P(n), a(n), Y(n))|P(n) = P].
\]

First, we show the limiting behavior of the sequence of \( \{P(n)\} \) in the following proposition.

**Proposition 2:** As the step size \( b \to 0 \), the sequence of the piecewise-constant interpolation of \( \{P(n)\} \) defined by \( \hat{P}(t) = P(n) \), for \( t \in [nb, (n + 1)b) \), converges weakly to the solution \( X^*(t) \) of the ODE defined by
\[
\frac{dX(t)}{dt} = f(X(t)), \quad X(0) = \hat{P}(0) = P(0) \tag{18}
\]
where \( P(0) \) is the initial channel selection probability matrix.

**Proof:** The result is a direct application of [Theorem 3.1].

Proposition 2 indicates that for sufficiently small step size, the long-term behavior of the sequence \( \{P(n)\} \) follows the trajectory \( X^*(t) \) of the ODE in (18). This allows us to use the stability properties of ODE to analyze the algorithm. The following proposition captures the relationship between the stable stationary points of the ODE and the Nash equilibria of \( \mathcal{S}_p \).

**Proposition 3:** All the stable stationary points of the ODE in (18) are the Nash equilibria of the game \( \mathcal{S}_p \). All the pure-strategy Nash equilibria of \( \mathcal{S}_p \) are the asymptotically-stable stationary points of the ODE in (18).

**Proof:** The result follows immediately from [16, Theorem 3.2].

Based on Propositions 2 and 3, we now characterize the long term behavior of the channel selection algorithm in our proposed DALA policy.

**Proposition 4:** Assuming a sufficiently small step size \( b \), under our proposed DALA policy, the channel selection probability vector converges to a pure strategy NE of the game \( \mathcal{S}_p \).

**Proof:** See Appendix B.

**VI. DISCUSSIONS**

1) **Collision Model:** We have so far assumed a relatively simple collision model that if more than one SU transmit on a channel, then no one is successful. However, our proposed DALA policy does not rely on this assumption, and can be readily applied to other collision models. Depending on the physical (PHY) layer and/or medium access control (MAC) layer implementations, it is possible that there is a non-zero probability of packet reception when more than one SU transmits. In this case, whether an SU’s packet is successful or not is indicated by its acknowledgement \( \zeta(n) \), and our DALA policy can still be carried out based on this information as in Algorithm 1. What is affected is each SU’s throughput, since the throughput expression in (7) would be different. The formulated game is still applicable, but again with different reward expression in (11), and consequently, the expression of payoff (expected throughput) in (13). Their exact expressions depend on the specific PHY/MAC implementations (i.e., collision model). Whether or not the resulting game is still an exact potential game needs to be evaluated in a case-by-case manner, since the potential function depends on the expression of payoff. Under some common MAC protocols such as carrier sensing multiple access (CSMA), the game can still be shown to be an exact potential game. In this case, the convergence result in Proposition 4 will still hold. Again, for other collision models, it should be evaluated in a case-by-case manner.
TABLE II
C A S E S O F M E A N C H A N N E L A V A I L A B I L I T Y \( \theta \) I N S I M U L A T I O N S

<table>
<thead>
<tr>
<th>( N )</th>
<th>( M )</th>
<th>Case</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>4</td>
<td>1</td>
<td>{0.1, 0.2, ..., 0.9}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>{0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>{0.51, 0.52, ..., 0.59}</td>
</tr>
</tbody>
</table>

2) Dynamically Changing Primary Loads: In developing our proposed DALA policy, we have assumed that the primary channel mean availabilities \( \theta_i \)'s are fixed. This is the underlying assumption for estimating \( \theta_i \)'s based on each SU's sensing history. Our policy could be generalized to a slowly changing dynamic environment where \( \theta_i \)'s change over time slowly. In this case, the learning process may be able to track the change in \( \theta_i \)'s. If the change is sufficiently slow, then the convergence results given in Section V would still hold within each time period of a new set of \( \theta_i \)'s.

3) SU Population: Our proposed DALA policy is distributed in terms of learning and access. However, each SU requires to know the SU population \( M \). This is also the case in several existing distributed access policies [4]–[6], such as \( \text{RAND} \) and DLF. For an infrastructured secondary network, the population \( M \) can be shared among SUs easily through broadcasting. However, for an ad hoc secondary network, \( M \) would need to be estimated and tracked at each SU. In [7], we have investigated this problem, and proposed an algorithm to estimate the unknown (and possibly changing) SU population. It is also shown that inaccurate estimation of \( M \) results in linear growth of regret (i.e., reduced throughput). Moreover, underestimation of \( M \) incurs more significant throughput loss than overestimation does.

4) Time Synchronization: In this work, we assume a slotted network and perfect synchronization, where SUs are slot-synchronized to the primary slots for sensing and access, and sensing and data transmission among SUs can be perfectly differentiated. Time synchronization is an important subject for research in cognitive radio network designs. It is especially challenging for an ad hoc secondary network instead of an infrastructured secondary network. Some distributed synchronization protocols have been proposed in literature [36]–[38]. In general, imperfect synchronization in practice will cause missing spectrum opportunity or collision among SUs, resulting in reduced throughput and/or increased interference to the primary network.

VII. Simulation Results

In this section, we study the performance of our proposed DALA policy in Algorithm 1. We assume a cognitive radio network with \( M = 4 \) SUs independently searching among \( N = 9 \) primary channels. We assume each SU always has data to send at every time slot. All simulations are performed with 200 Monte Carlo runs.

At each time slot \( n \), the channel availability state \( X_i(n) \) for channel \( i \) follows an i.i.d Bernoulli process over time with mean \( \theta_i \), and is independent from that of other channels. Table II lists

\[\text{Table II: Cases of Mean Channel Availability } \theta \text{ in Simulations}\]

10In this case, for a better tracking result, we could modify the sample mean estimate in (2) to some other tracking methods, such as moving average.

Fig. 1. Channel selection probability vs. time slot \( n \) for SU 1 (\( M = 4, N = 9 \), case 1 for \( \theta \), \( b = 0.01 \)).

Fig. 2. Channel selection probability vs. time slot \( n \) for SU 2 (\( M = 4, N = 9 \), case 1 for \( \theta \), \( b = 0.01 \)).

In implementing Algorithm 1, we use a threshold \( p_{th} \) to check the convergence of the channel selection probability vector \( p^i(n) \) to a unit vector (i.e., pure strategy): If \( p_{th} < p_{th} \), for some \( i^o \), then we set \( p_{i^o} = 1 \) and \( p_i = 0 \) for \( i \neq i^o \). In the simulation, we set \( p_{th} = 0.9 \).

A. Convergence Behavior of Channel Selection Probabilities

We first show in Figs. 1 to 4 the convergence behavior of channel selection probability \( p^i(n) \) over time slot \( n \) for each SU \( j \), under Algorithm 1. Case 1 is used for \( \theta \). As it can be seen from Fig. 1, for SU 1, the channel selection probability on channel 6 converges to 1, while that on the rest of channels converges to 0, i.e. a pure strategy with \( p^i(n) \rightarrow e_6 \) as all the cases of the mean channel availability vector \( \theta \) we consider in the simulation. These cases represents different types of traffic load distribution across the primary channels. Case 1 represents a scenario of dissimilar channel loads, where the mean channel availabilities are evenly spread out. Case 2 represents a scenario where the average loads across different channels are random. Case 3 provides an example where the average loads on all channels are similar.

In implementing Algorithm 1, we use a threshold \( p_{th} \) to check the convergence of the channel selection probability vector \( p^i(n) \) to a unit vector (i.e., pure strategy): If \( p_{th} > p_{th} \), for some \( i^o \), then we set \( p_{i^o} = 1 \) and \( p_i = 0 \) for \( i \neq i^o \). In the simulation, we set \( p_{th} = 0.9 \).
Fig. 3. Channel selection probability vs. time slot \(n\) for SU 3 \((M = 4, N = 9, \text{ case 1 for } \theta, b = 0.01)\).

Fig. 4. Channel selection probability vs. time slot \(n\) for SU 4 \((M = 4, N = 9, \text{ case 1 for } \theta, b = 0.01)\).

\(n\) becomes large. Thus, SU 1 asymptotically selects channel 6 with probability approaching to 1. Similarly, Fig. 2 shows that the channel selection probability vector for SU 2 converges to \(p_2(n) \rightarrow e_7\) and SU 2 asymptotically selects channel 7 with probability approaching to 1. Also Figs. 3 and 4 show that, for SUs 3 and 4, their channel selection probability vectors converge to \(p_3(n) \rightarrow e_8\) and \(p_4(n) \rightarrow e_9\), respectively, as \(n\) becomes large. We see that all SUs asymptotically select a specific channel orthogonal to other SUs to access. Furthermore, the four selected channels are among the \(M\)-best channels, with probability approaches to 1, and the channel selections among SUs are orthogonalized.

B. Comparison With Existing Access Policies

We now compare the performance of our proposed policy with three existing learning and access policies: \(\rho^{\text{RAND}}\) [5], DLF [6], and another SLA-based channel selection algorithm proposed in [10]11.

As explained in Section II, regret \(R(n)\) in (1) is a common metric used to measures the gap of the achieved throughput under a given policy to the optimal throughput over time slot \(n\). In existing policies such as \(\rho^{\text{RAND}}\) and DLF, it is shown that the regret has a logarithm growth rate over \(n\), i.e., \(R(n) = O(\log n)\), which is order optimal [11]. The performance difference among existing policies is the leading constant of the growth rate. To evaluate the performance of our proposed policy, we use the normalized regret, \(R(n)/\log n\), normalized against the regret growth rate. It reflects the leading constant of the growth rate. A lower value of the normalized regret indicates a better throughput performance.12

Fig. 5 shows the normalized regret over time slot \(n\) under the aforementioned four access policies. Case 1 for \(\theta\) is used, representing the case with dissimilar mean channel availabilities across primary channels.13 In Fig. 6, the same comparison is

\begin{table}[h]
\centering
\caption{Channel Selected By Each SU (\(\theta\): Case 1)}
\begin{tabular}{|c|c|c|}
\hline
\(N\) & \(M\) & SU & Selected Channel \\
\hline
9 & 4 & 1 & 6 \\
 & & 2 & 7 \\
 & & 3 & 8 \\
 & & 4 & 9 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Channel Selected By Each SU (\(\theta\): Case 2)}
\begin{tabular}{|c|c|c|}
\hline
\(N\) & \(M\) & SU & Selected Channel \\
\hline
9 & 4 & 1 & 5 \\
 & & 2 & 9 \\
 & & 3 & 6 \\
 & & 4 & 8 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Channel Selected By Each SU (\(\theta\): Case 3)}
\begin{tabular}{|c|c|c|}
\hline
\(N\) & \(M\) & SU & Selected Channel \\
\hline
9 & 4 & 1 & 6 \\
 & & 2 & 8 \\
 & & 3 & 9 \\
 & & 4 & 7 \\
\hline
\end{tabular}
\end{table}

11The policy proposed in [10] does not take the channel availability into account, thus treats every channel in the same manner.

12The throughput performance comparison can be directly converted from the regret comparison. Thus, we only show regret comparison over time.

13In Fig. 5, we plot the normalized regret of the secondary network over time slot \(n\) under our proposed policy. Compared with Figs. 1-4, the convergence is much slower. Note that Figs. 1-4 show the channel selection probability over time slot \(n\) for each SU, respectively. These are two different quantities and have different convergence rates. Channel selection probability mainly is affected by the learning process through collision, while the regret is mainly affected by the learning process of both \(\theta_i\)’s and collision.
performed for $\theta$ in Case 2 where the mean channel availabilities are set randomly. We also examine the case where channels are similar in their mean availabilities, i.e., $\theta$ in Case 3, in Fig. 7.

From Figs. 5 to 7, we see that, as discussed in Section III-D, the relative performance of $\rho^{\text{ RAND}}$ versus DLF depends on the type of distribution in $\theta$. Nonetheless, our proposed policy significantly outperforms these two policies in all three types of $\theta$ values. Furthermore, our proposed policy significantly outperforms the policy in [10] in all cases except for Case 3 with similar mean channel availabilities, where the policy in [10] is slightly better. In [10], SUs do not distinguish different primary channels and can select any one of those channels. When $\theta$’s are dissimilar, using our policy, SUs avoid those channels with low availability and thus result in improved throughput. However, when $\theta$’s are all similar, selecting channels other than the $M$-best channels will result in negligible throughput loss. On the contrary, selecting from more channels than $M$-best channels will result in reduced collision among SUs, and thus improve the throughput. Thus, in this case, the policy in [10] performs slightly better than our policy.

In summary, the simulation results show the effectiveness of our proposed policy for different $\theta$ distributions, as compared with the existing access policies. They demonstrate that our proposed policy automatically adapts the channel selection process based on $\theta$ distribution, and thus works well for a wide range of mean availability distribution of the primary channels.

VIII. Conclusion

In this work, we considered the problem of decentralized online learning and channel access in a cognitive radio network through a game theoretic approach. First, we proposed an adaptive learning and access policy which can effectively respond to different load conditions (i.e., mean availabilities) across primary channels. Specifically, each SU probabilistically selects one of its estimated $M$-best channels to access, and the SU updates the channel selection probability based on collision events. Our proposed adaptive policy consists of two underlying distributed learning algorithms, one is UCB-based learning from sensing history on the primary channel availability, and the other is SLA-based learning from collision history on channel selections among SUs to avoid further collision. Next, we formulate the distributed channel selection and access problem as a game which is shown to be an exact potential game, and therefore can at least have a pure strategy NE point. We then proved that under our proposed learning and access policy, the channel selection probabilities converge towards a pure strategy NE point of the game. Numerical results showed the effectiveness of our proposed adaptive policy in various distributions of mean availabilities across primary channels, as compared with other existing policies.

We would like to point out that we have focused on designing a learning and access policy which can perform well for a wide range of $\theta$ distributions for the primary channels. We have not considered fairness among SUs in this work. Ensuring certain fairness among SUs in a distributed protocol design is particularly challenging. How to include fairness in both learning and access processes is an important subject for future research.

14Fairness is considered in DLF policy; however, additional information is needed at each SU. Besides knowing $M$, each SU is assigned a unique sequential ID, which is used for channel selection in a round robin fashion to ensure fairness among SUs.
\section*{Appendix A}

\textbf{Proof of Proposition 1}

\textit{Proof:} We define a potential function known as Rosenthal’s potential function \cite{39} for our game $\mathcal{G}_p$ as follows:
\[
\mathcal{P}(a_j, a_{-j}; n) \overset{\Delta}{=} \sum_{i=1}^{N} \sum_{k=1}^{m_i(n)} \psi_i(k),
\]
where $\psi_i(k)$ is defined in (14).

Let assume $a_j(n)$ and $\tilde{a}_j(n)$ as two different actions which SU $j$ may take from the set of possible actions, i.e., $a_j(n), \tilde{a}_j(n) \in A_j(n)$ and $a_j(n) \neq \tilde{a}_j(n)$. We define $\hat{\psi}_M(n)$ as
\[
\hat{\psi}_M(n) \overset{\Delta}{=} \psi_M(n)\{a_j(n), \tilde{a}_j(n)\},
\]
Then, the potential function in (19) can be rewritten by
\[
\mathcal{P}(a_j, a_{-j}; n) = \sum_{i \in \hat{\psi}_M(n)} \sum_{k=1}^{m_i(n)} \psi_i(k) = \sum_{i \in \hat{\psi}_M(n)} \sum_{k=1}^{m_i(n)} \psi_i(k) + \sum_{k=1}^{m_{a_j(n)}(n) - 1} \psi_{a_j}(k) + \sum_{k=1}^{m_{\tilde{a}_j(n)}(n) + 1} \psi_{\tilde{a}_j}(k).
\]

For SU $j$ changes its channel access to $\tilde{a}_j(n) \in A_j(n)$, the potential function is given by
\[
\mathcal{P}(\tilde{a}_j, a_{-j}; n) = \sum_{i \in \hat{\psi}_M(n)} \sum_{k=1}^{m_i(n)} \psi_i(k) = \sum_{i \in \hat{\psi}_M(n)} \sum_{k=1}^{m_i(n)} \psi_i(k) + \sum_{k=1}^{m_{a_j(n)}(n) - 1} \psi_{a_j}(k) + \sum_{k=1}^{m_{\tilde{a}_j(n)}(n) + 1} \psi_{\tilde{a}_j}(k).
\]

Thus, when SU $j$ changes its channel access from $a_j(n)$ to $\tilde{a}_j(n)$, the deviation in the potential function is given by
\[
\mathcal{P}(\tilde{a}_j, a_{-j}; n) - \mathcal{P}(a_j, a_{-j}; n) = \psi_{\tilde{a}_j}(m_{\tilde{a}_j}(n) + 1) - \psi_{a_j}(m_{a_j}(n)).
\]

For SU $j$, the change in its payoff by switching from $a_j(n)$ to $\tilde{a}_j(n)$ can be obtained by
\[
u_j(\tilde{a}_j, a_{-j}; n) - u_j(a_j, a_{-j}; n) = \psi_{\tilde{a}_j}(m_{\tilde{a}_j}(n) + 1) - \psi_{a_j}(m_{a_j}(n)).
\]

By (21) and (22), it follows that the property in (15) holds. Therefore, the game $\mathcal{G}_p$ is an exact potential game. \hfill \blacksquare

\section*{Appendix B}

\textbf{Proof of Proposition 4}

\textit{Proof:} Assume that SU $j$ chooses a pure strategy of selecting channel $i$, and any other SU $s \in \mathcal{S}, s \neq j$, takes a mixed strategy $p'_s(n)$. Let $P_{-j}(n) \overset{\Delta}{=} \{p'_s(n) : s \in \mathcal{S} \setminus \{j\}\}$ be the set of channel probability selection vectors of the SU $j$’s opponents at slot time $n$, and let $e_j$ be a unit vector with the $i$th entry being 1 and the rest 0’s. Then, the expected throughput of SU $j$ at slot time $n$, taken over all other SUs’ actions $a_{-j}$, and denoted by $\bar{u}_j(e_j, P_{-j}; n)$, is given by
\[
\bar{u}_j(e_j, P_{-j}; n) \overset{\Delta}{=} \mathbb{E}[u_j(i, a_{-j}; n)] = \sum_{a_j(n) \in A_j(n)} \sum_{i \in \mathcal{S} \setminus \{j\}} a_j(n) \in A_j(n) \mathbb{E}[u_j(i, a_{-j}; n)]
\]
Since the channel selection $a_j(n)$ and $a_{-j}$ are random variables following $p'_j(n)$ and $P_{-j}(n)$, the potential function $\mathcal{P}(a_j, a_{-j}; n)$ in (19) is random. We define $X(p'_j, P_{-j}; n) : P \rightarrow R$ as the expected value of $\mathcal{P}(a_j, a_{-j}; n)$ by SU $j$
\[
X(p'_j, P_{-j}; n) \overset{\Delta}{=} \mathbb{E}[\mathcal{P}(a_j, a_{-j}; n)|p'_j, P_{-j}].
\]
Then, for SU $j$ taking a pure strategy $p'_j(n) = e_i$, we have $a_j = i$ and
\[
X(e_i, P_{-j}; n) = \mathbb{E}[\mathcal{P}(i, a_{-j}; n)|p'_j, P_{-j}],
\]

From (15) and (23), for SU $j$ changes its selection from channel $i$ to $i'$, we have
\[
X(e_i, P_{-j}; n) - X(e_{i'}, P_{-j}; n) = \bar{u}_j(e_i, P_{-j}; n) - \bar{u}_j(e_{i'}, P_{-j}; n).
\]

To prove our claim, we use the result in [10] which is stated below.

\textit{Theorem 1:} [10, Theorem 5] Suppose that there is a non-negative function $X(p'_j, P_{-j}; n) : P \rightarrow R$ for some positive constant $c > 0$ such that
\[
X(e_i, P_{-j}; n) - X(e_{i'}, P_{-j}; n) = c[\bar{u}_j(e_i, P_{-j}; n) - \bar{u}_j(e_{i'}, P_{-j}; n)], \forall j, i, i', P.
\]

Then, the SLA-based algorithm converges to a pure strategy NE of a game.

Based on the result in Theorem 1, from (26), it follows that our proposed algorithm converges to pure strategy NE points of the game $\mathcal{G}_p$. \hfill \blacksquare

\section*{References}


Ali Grami (M’86–SM’06) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, the M.Eng. degree from McGill University, Montreal, QC, Canada, and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, all in electrical engineering. Following his graduation, he joined Nortel Networks, where he was involved in the research, design, and development of North America’s first digital cellular mobile system. He then joined Telesat Canada, where he was the Lead Researcher and the Principal Designer of Canada’s Anik-F2 Ka-band system, the world’s first broadband access satellite system and the first satellite to successfully commercialize the Ka-band technology. While he was with the industry, he taught at the University of Ottawa, Ottawa, ON, Canada, and Concordia University, Montreal, QC, Canada. In 2003, he, as a founding Faculty Member, joined the University of Ontario Institute of Technology (UOIT), where he has led the development of the B.Eng., M.Eng., and Ph.D. programs in electrical and computer engineering. He has been the recipient of the United Nations TOKTEN award. He is also the author of the undergraduate-level textbook *Introduction to Digital Communications* (Elsevier, 2016).